A
Project Report
on

HEART DISEASE PREDICTION MODEL

Submitted in partial fulfillment of the requirements
for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering (Artificial Intelligence)

by
Ashok (2200971529002)
Latif Ahmed (2200971529003)
Shivendra Tiwari (2200971529005)

Under the Supervision of
Dr. Sunil Kumar

Galgotias College of Engineering & Technology
Greater Noida, Uttar Pradesh
India-201306
Affiliated to

Dr. A.P.J. Abdul Kalam Technical University
Lucknow, Uttar Pradesh,
India-226031
December, 2023

GALGOTIAS COLLEGE OF ENGINEERING & TECHNOLOGY
GREATERNOIDA,UTTARPRADESH, INDIA- 201306.

CERTIFICATE

This is to certify that the project report entitled "HEART DISEASE PREDICTION MODEL" submitted by Ashok - Roll. No. 2200971529002, Latif Ahmed - Roll. No. 2200971529003, Shivendra Tiwari - Roll. No. 2200971529005 to the Galgotias College of Engineering & Technology, Greater Noida, Utter Pradesh, affiliated to Dr. A.P.J. Abdul Kalam Technical University Lucknow, Uttar Pradeshinpartial fulfillment for the award of Degree of Bachelor of Technology in Computer Science & Engineering is a bonafide record of the project work carried out by the under my supervision during the year 2023-2024.

Dr. Sunil Kumar
Associate Professor
Dept.of CSE

Dr. Vishnu Sharma
Professor and Head
Dept.of CSE

GALGOTIAS COLLEGE OF ENGINEERING & TECHNOLOGY
GREATERNOIDA,UTTARPRADESH, INDIA- 201306.

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend my sincere thanks to all of them.

We are highly indebted to **Dr. Sunil Kumar** for **his** guidance and constant supervision. Also, we are highly thankful to them for providing necessary information regarding the project & also for their support in completing the project.

We are extremely indebted to Dr. Vishnu Sharma, HOD, Department of Computer Science and Engineering, GCET and Dr. Jaya Sinha / Mr. Manish Kumar Sharma, Project Coordinator, Department of Computer Science and Engineering, GCET for their valuable suggestions and constant support throughout my project tenure. We would also like to express our sincere thanks to all faculty and staff members of Department of Computer Science and Engineering, GCET for their support in completing this project on time.

We also express gratitude towards our parents for their kind co-operation and encouragement which helped me in completion of this project. Our thanks and appreciations also go to ourfriends in developing the project and all the people who have willingly helped me out with their abilities.

Ashok
Latif Ahmed
Shivendra Tiwari

ABSTRACT

The diagnosis of heart disease in most cases depends on a complex combination of clinical and pathological data. Because of this complexity, there exists a significant amount of interest among clinical professionals and researchers regarding the efficient and accurate prediction of heart disease. In this paper, we develop a heart disease predict system that can assist medical professionals in predicting heart disease status based on the clinical data of patients. Our approaches include three steps. Firstly, we select 13 important clinical features, i.e., age, sex, chest pain type, trestbps, cholesterol, fasting blood sugar, resting ecg, max heart rate, exercise induced angina, old peak, slope, number of vessels colored, and thal. Secondly, we develop an artificial neural network algorithm for classifying heart disease based on these clinical features. The accuracy of prediction is near 80%. Finally, we develop a user-friendly heart disease predict system (HDPS). The HDPS system will be consisted of multiple features, including input clinical data section, ROC curve display section, and prediction performance display section (execute time, accuracy, sensitivity, specificity, and predict result). Our approaches are effective in predicting the heart disease of a patient. The HDPS system developed in this study is a novel approach that can be used in the classification of heart disease.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

According to the World Health Organization, every year 12 million deaths occur worldwide due to Heart Disease. The load of cardiovascular disease is rapidly increasing all over the world from the past few years. Many researches have been conducted in attempt to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. Heart Disease is even highlighted as a silent killer which leads to the death of the person without obvious symptoms. The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications. This project aims to predict future Heart Disease by analyzing data of patients which classifies whether they have heart disease or not using machine-learning algorithms.

1.1 PROBLEM DEFINITION

The major challenge in heart disease is its detection. There are instruments available which can predict heart disease but either they are expensive or are not efficient to calculate chance of heart disease in human. Early detection of cardiac diseases can decrease the mortality rate and overall complications. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time and expertise. Since we have a good amount of data in today's world, we can use various machine learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

1.2 MOTIVATION

Machine learning techniques have been around us and has been compared and used for analysis for many kinds of data science applications. The major motivation behind this research-based project was to explore the feature selection methods, data preparation and processing behind the training models in the machine learning. With first hand models and libraries, the challenge we face today is data where beside their abundance, and our cooked models, the accuracy we see during training, testing and actual validation has a higher variance. Hence this project is carried out with the motivation to explore

behind the models, and further implement Logistic Regression model to train the obtained data. Furthermore, as the whole machine learning is motivated to develop an appropriate computer-based system and decision support that can aid to early detection of heart disease, in this project we have developed a model which classifies if patient will have heart disease in ten years or not based on various features (i.e. potential risk factors that can cause heart disease) using logistic regression. Hence, the early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine.

1.3 OBJECTIVES

The main objective of developing this project are:

1. To develop machine learning model to predict future possibility of heart disease by implementing Logistic Regression.
2. To determine significant risk factors based on medical dataset which may lead to heart disease.
3. To analyze feature selection methods and understand their working principle.

Chapter 2

LITERATURE REVIEW

In recent years, the healthcare industry has seen a significant advancement in the field of data mining and machine learning. These techniques have been widely adopted and have demonstrated efficacy in various healthcare applications, particularly in the field of medical cardiology. The rapid accumulation of medical data has presented researchers with an unprecedented opportunity to develop and test new algorithms in this field. Heart disease remains a leading cause of mortality in developing nations [12,13,14,15,16], and identifying risk factors and early signs of the disease has become an important area of research. The utilization of data mining and machine learning techniques in this field can potentially aid in the early detection and prevention of heart disease.

The purpose of the study described by Narain et al. (2016) is to create an innovative machine-learning-based cardiovascular disease (CVD) prediction system in order to increase the precision of the widely used Framingham risk score (FRS). With the help of data from 689 individuals who had symptoms of CVD and a validation dataset from the Framingham research, the proposed system—which uses a quantum neural network to learn and recognize patterns of CVD—was experimentally validated and compared with the FRS. The suggested system's accuracy in forecasting CVD risk was determined to be 98.57%, which is much greater than the FRS's accuracy of 19.22% and other existing techniques. According to the study's findings, the suggested approach could be a useful tool for doctors in forecasting CVD risk, assisting in the creation of better treatment plans, and facilitating early diagnosis.

In a study conducted by Shah et al. (2020), the authors aimed to develop a model for predicting cardiovascular disease using machine learning techniques. The data used for this purpose were obtained from the Cleveland heart disease dataset, which consisted of 303 instances and 17 attributes, and were sourced from the UCI machine learning repository. The authors employed a variety of supervised classification methods, including naive Bayes, decision tree, random forest, and k-nearest neighbor (KKN). The results of the study indicated that the KKN model exhibited the highest level of accuracy, at 90.8%. The study highlights the potential utility of machine learning techniques in predicting cardiovascular disease, and emphasizes the importance of selecting appropriate models and techniques to achieve optimal results.

In a study by Drod et al. (2022), the objective was to use machine learning (ML) techniques to identify the most significant risk variables for cardiovascular disease (CVD)

in patients with metabolic-associated fatty liver disease (MAFLD). Blood biochemical analysis and subclinical atherosclerosis assessment were performed on 191 MAFLD patients. A model to identify those with the highest risk of CVD was built using ML approaches, such as multiple logistic regression classifier, univariate feature ranking, and principal component analysis (PCA). According to the study, hypercholesterolemia, plaque scores, and duration of diabetes were the most crucial clinical characteristics. The ML technique performed well, correctly identifying 40/47 (85.11%) high-risk patients and 114/144 (79.17%) low-risk patients with an AUC of 0.87. According to the study's findings, an ML method is useful for detecting MAFLD patients with widespread CVD based on simple patient criteria.

In a study published by Alotalibi (2019), the author aimed to investigate the utility of machine learning (ML) techniques for predicting heart failure disease. The study utilized a dataset from the Cleveland Clinic Foundation, and implemented various ML algorithms, such as decision tree, logistic regression, random forest, naive Bayes, and support vector machine (SVM), to develop prediction models. A 10-fold cross-validation approach was employed during the model development process. The results indicated that the decision tree algorithm achieved the highest accuracy in predicting heart disease, with a rate of 93.19%, followed by the SVM algorithm at 92.30%. This study provides insight into the potential of ML techniques as an effective tool for predicting heart failure disease and highlights the decision tree algorithm as a potential option for future research.

Through a comparison of multiple algorithms, Hasan and Bao (2020) carried out a study with the main objective of identifying the most efficient feature selection approach for anticipating cardiovascular illness. The three well-known feature selection methods (filter, wrapper, and embedding) were first taken into account, and then a feature subset was recovered from these three algorithms using a Boolean process-based common "True" condition. This technique involved retrieving feature subsets in two stages. A number of models, including random forest, support vector classifier, k-nearest neighbors, naive Bayes, and XGBoost, were taken into account in order to justify the comparative accuracy and identify the best predictive analytics. As a standard for comparison with all features, the artificial neural network (ANN) was used. The findings demonstrated that the most accurate prediction results for cardiovascular illness were provided by the XGBoost classifier coupled with the wrapper technique. XGBoost delivered an accuracy of 73.74%, followed by SVC with 73.18% and ANN with 73.20%.

The primary drawback of the prior research is its limited dataset, resulting in a high risk of overfitting. The models developed may not be appropriate for large datasets. In contrast, we utilized a cardiovascular disease dataset consisting of 70,000 patients and 11 features, thereby reducing the chance of overfitting. Table 1 presents a concise review of cardiovascular disease prediction studies performed on large datasets, further reinforcing the effectiveness of using a substantial dataset.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Table 2.1: Different approaches and their accuracy on different data-sets

S.No	Novel Approach	Best Accuracy	Dataset
1.	Stacking of KNN, random forest, and SVM outputs with logistic regression as the metaclassifier	75.1% (stacked model)	Kaggle cardiovascular disease dataset (70,000 patients, 12 attributes)
2	Random forest -Naive Bayes -Logistic regression -KNN	70%	Kaggle cardiovascular disease dataset (70,000 patients, 12 attributes)
3	Decision tree	72.77% (decision tree)	Kaggle cardiovascular disease dataset (70,000 patients, 12 attributes)
4	Repeated random with random forest	89.01%(random forest classifier)	UCI cardiovascular dataset (303 patients, 14 attributes)

Chapter 3

PROBLEM FORMULATION

Creating a heart disease prediction model involves several crucial steps, with problem formulation being fundamental. The aim is to develop a model that accurately forecasts the likelihood of an individual developing heart disease based on various factors.

Firstly, it's essential to define the problem scope and objectives clearly. This involves determining the target population for the model—whether it's the general public or a specific demographic, such as individuals with certain risk factors or medical histories. Identifying the specific types of heart diseases to predict and the factors to consider (like age, gender, blood pressure, cholesterol levels, family history, lifestyle habits, etc.) is crucial.

Next, data collection and preprocessing play a pivotal role. Gathering comprehensive and diverse datasets containing relevant patient information is necessary. Data preprocessing involves cleaning the data, handling missing values, normalizing or scaling features, and encoding categorical variables. Additionally, ensuring data privacy and compliance with ethical standards is paramount.

Model selection and evaluation follow this stage. Choosing the appropriate machine learning algorithms (like logistic regression, random forests, support vector machines) and validating their performance using techniques like cross-validation, ROC curves, precision-recall curves, and metrics like accuracy, sensitivity, specificity, and F1 score are critical.

Lastly, deploying the model in a real-world setting involves considering its scalability, interpretability, and usability. Ensuring that the model can be integrated into healthcare systems securely and providing actionable insights to healthcare professionals is essential for its practical application.

In summary, problem formulation in heart disease prediction models necessitates a thorough understanding of the problem scope, data collection, preprocessing, feature engineering, model selection, evaluation, and real-world deployment considerations to build a robust and effective predictive tool for identifying individuals at risk of heart diseases.

Chapter 4

METHODOLOGY

4.1 DESCRIPTION OF THE DATASET

The dataset used for this research purpose was the Public Health Dataset and it is dating from 1988 and consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V. It contains 76 attributes, including the predicted attribute, but all published experiments refer to using a subset of 14 of them[1]. Now the attributes which are used in this research purpose are described as follows and for what they are used or resemble:

- Age—age of patient in years,
- sex—(1=male; 0=female)
- Cp—chest pain type.
- Trestbps—resting blood pressure (in mm Hg on admission to the hospital). The normal range is 120/80 (if you have a normal blood pressure reading, it is fine, but if it is a little higher than it should be, you should try to lower it. Make healthy changes to your lifestyle).
- Chol—serum cholesterol shows the amount of triglycerides present. Triglycerides are another lipid that can be measured in the blood. It should be less than 170mg/dL (may differ in different Labs).
- Fbs—fasting blood sugar larger than 120mg/dl (1 true). Less than 100mg/dL (5.6mmol/L) is normal, and 100 to 125mg/dL (5.6 to 6.9mmol/L) is considered prediabetes.
- Restecg—resting electrocardiographic results.
- Thalach—maximum heart rate achieved. The maximum heart rate is 220 minus your age.
- Exang—exercise-induced angina (1 yes). Angina is a type of chest pain caused by reduced blood flow to the heart. Angina is a symptom of coronary artery disease.
- Oldpeak—ST depression induced by exercise relative to rest.

- Slope—the slope of the peak exercise ST segment.
- Ca—number of major vessels (0–3) colored by fluoroscopy.
- Thal—no explanation provided, but probably thalassemia (3 normal; 6 fixed defects; 7 reversible defects).
- Target (T)—no disease=0 and disease=1, (angiographic disease status).

4.2 PROPOSED WORK

This project will involve these steps-

1. Acquiring of data
2. Filtering of data
3. Transforming of data
4. Data Analysis (Exploring the data)
5. Splitting the data for test model
6. Algorithm Implementation
7. Final Model Implementation

- **Acquisition of data-**

  This process involves acquiring the data on which we will be building our model. In this model we will use the data of Framingham Heart Study (FHS) established in 1948.

- **Filtering the data-**

  (a) The dataset does not have any null values. But many outliers needed to be handled properly, and also the dataset is not properly distributed. Two approaches were used.

  (b) One witout outliers and feature selection process and directly applying the data to the machine learning algorithms, and the results which were achieved were not promising.

  (c) But after using the normal distribution of dataset for overcoming the overfitting problem and then applying Isolation Forest for the outlier's detection, the results achieved are quite promising.

  (d) Various plotting techniques were used for checking the skewness of the data, outlier detection, and the distribution of the data. All these preprocessing

techniques play an important role when passing the data for classification or prediction purposes[2].

- **Transforming the data-**

  In this step, we will be transforming the data. Example- If there are any missing values, will replace them with mean value, etc.

- **Data Analysis-**

  This process involves data cleaning, data statistics and getting insights from the dataset.

  (a) The distribution of the data plays an important role when the prediction or classification of a problem is to be done.

  (b) We see that the heart disease occurred 48.7% of the time in the dataset, whilst 51.3% was the no heart disease.

  (c) So, we need to balance the dataset or otherwise it might get overfit. This will help the model to find a pattern in the dataset that contributes to heart disease[3],

- **Splitting the data for test model-**

  In this process, we be splitting the data for training our model, so we can implement our algorithm on it.

- **Algorithm Implementation-**

  This involves four machine learning algorithms which will result in performance metrics of the model.

  Logistic Regression - Logistic Regression is a supervised classification algorithm. It is a predictive analysis algorithm based on the concept of probability. It measures the relationship between the dependent variable (target) and the one or more independent variables (risk factors) by estimating probabilities using underlying logistic function (sigmoid function). Sigmoid function is used as a cost function to limit the hypothesis of logistic regression between 0 and 1 (squashing).

  Logistic Regression relies highly on the proper presentation of data. So, to make the model more powerful, important features from the available data set are selected using Backward elimination and recursive elimination techniques[4].

- **Model Implementation-**

  The well-doing algorithm is implemented in the model and checking results with the real-time data.

Chapter 5

SYSTEM DESIGN

Designing a system for heart disease prediction involves several components and considerations. Below is a high-level overview of the system design for a heart disease prediction project. Keep in mind that this is a simplified example, and actual implementations may vary based on specific requirements, data availability, and the chosen technology stack.

5.1 SYSTEM COMPONENTS:

- **Data Collection:**

  Collect relevant data for heart disease prediction. This can include medical records, patient history, lifestyle information, and diagnostic test results. Ensure the data is diverse and representative to improve the model's generalization.

- **Data processing:**

  Clean and preprocess the collected data to handle missing values, outliers, and inconsistencies. Normalize or standardize numerical features. Encode categorical variables. Split the dataset into training and testing sets.

- **Feature Engineering:**

  Extract relevant features from the data. Consider creating new features that might enhance the predictive power of the model.

- **Machine Learning Model:**

  Choose an appropriate machine learning algorithm for heart disease prediction. Common choices include logistic regression, decision trees, random forests, support vector machines, or neural networks. Train the model using the preprocessed data. Fine-tune hyperparameters to improve performance.

- **Model Evaluation:**

  Evaluate the model's performance using metrics such as accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC-ROC). Use cross-validation to ensure the model's robustness.

5.2 USE-CASE DIAGRAM

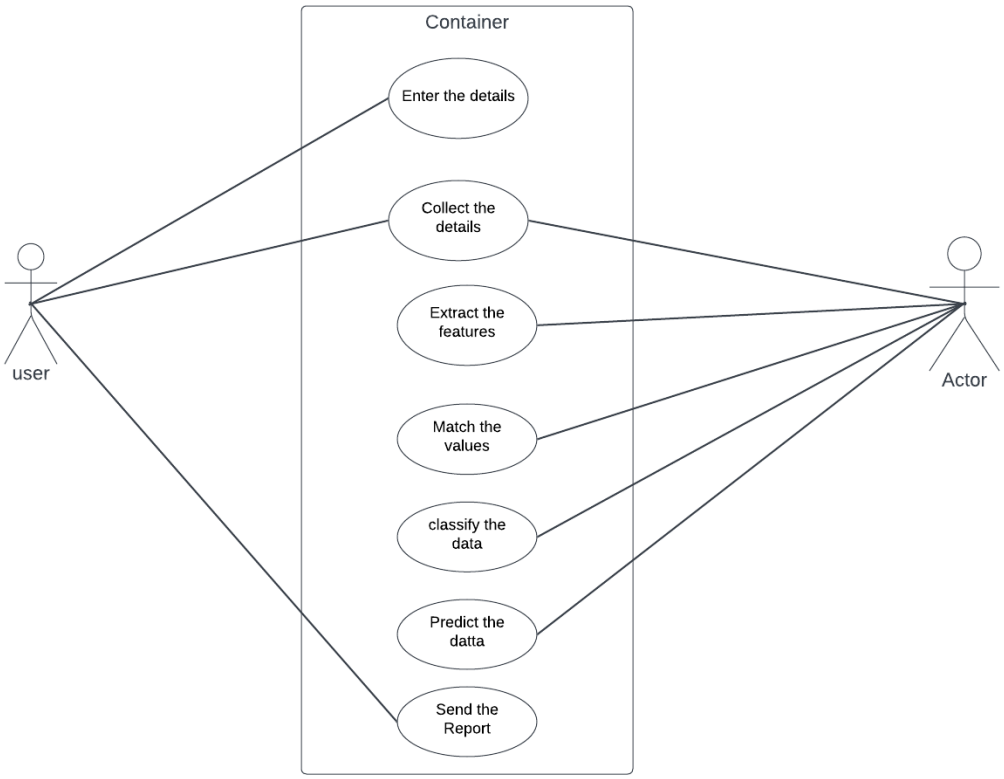

Figure 5.1: Use-case diagram of model

5.3 TECHNOLOGY STACK:

- **Programming Languages:**

  Python for data preprocessing, feature engineering, and model training.

- **Libraries/Frameworks:**

  Numpy, Pandas, Seaboars as sms or Spicy .

  Flask or Django for building the web application.

- **Database:**

  Use a relational database (e.g., PostgreSQL) to store patient data securely.

Chapter 6

IMPLEMENTATION

6.1 IMPORTING LIBRARIES

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy as sc
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

6.2 DATA COLLECTION AND PROCESSING

```
# loading the csv data
heart_data = pd.read_csv('/content/heart.csv')
```

```
# print first 5 rows of the dataset
heart_data.head()
```

```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   52    1   0       125   212    0        1      168      0      1.0      2
1   53    1   0       140   203    1        0      155      1      3.1      0
2   70    1   0       145   174    0        1      125      1      2.6      0
3   61    1   0       148   203    0        1      161      0      0.0      2
4   62    0   0       138   294    1        1      106      0      1.9      1

   ca  thal  target
0   2     3       0
1   0     3       0
2   0     3       0
3   1     3       0
4   3     2       0
```

```
# print last 5 rows of the dataset
```

```
heart_data.tail()
```

```
      age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
1020   59    1   1       140   221    0        1      164      1      0.0
1021   60    1   0       125   258    0        0      141      1      2.8
1022   47    1   0       110   275    0        0      118      1      1.0
1023   50    0   0       110   254    0        0      159      0      0.0
1024   54    1   0       120   188    0        1      113      0      1.4

      slope  ca  thal  target
1020      2   0     2       1
1021      1   1     3       0
1022      1   1     2       0
1023      2   0     2       1
1024      1   1     3       0
```

```
# number of rows and columns in dataset
heart_data.shape
```

```
(1025, 14)
```

```
# getting some info from dataset
heart_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1025 non-null   int64
 1   sex       1025 non-null   int64
 2   cp        1025 non-null   int64
 3   trestbps  1025 non-null   int64
 4   chol      1025 non-null   int64
 5   fbs       1025 non-null   int64
 6   restecg   1025 non-null   int64
 7   thalach   1025 non-null   int64
 8   exang     1025 non-null   int64
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64
 11  ca        1025 non-null   int64
 12  thal      1025 non-null   int64
```

```
 13  target    1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

```
# checching for missing value
heart_data.isnull().sum()
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

```
# stastical measures of the data
pd.set_option('display.float_format', lambda x: '%.3f' % x)
heart_data.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
age	1025.000	54.434	9.072	29.000	48.000	56.000	61.000	77.000
sex	1025.000	0.696	0.460	0.000	0.000	1.000	1.000	1.000
cp	1025.000	0.942	1.030	0.000	0.000	1.000	2.000	3.000
trestbps	1025.000	131.612	17.517	94.000	120.000	130.000	140.000	200.000
chol	1025.000	246.000	51.593	126.000	211.000	240.000	275.000	564.000
fbs	1025.000	0.149	0.357	0.000	0.000	0.000	0.000	1.000
restecg	1025.000	0.530	0.528	0.000	0.000	1.000	1.000	2.000
thalach	1025.000	149.114	23.006	71.000	132.000	152.000	166.000	202.000
exang	1025.000	0.337	0.473	0.000	0.000	0.000	1.000	1.000
oldpeak	1025.000	1.072	1.175	0.000	0.000	0.800	1.800	6.200
slope	1025.000	1.385	0.618	0.000	1.000	1.000	2.000	2.000
ca	1025.000	0.754	1.031	0.000	0.000	0.000	1.000	4.000
thal	1025.000	2.324	0.621	0.000	2.000	2.000	3.000	3.000

```
target    1025.000    0.513  0.500    0.000    0.000    1.000    1.000    1.000
```

```
# checking the distribution of target variable
heart_data['target'].value_counts(normalize=True)
```

```
1    0.513
0    0.487
Name: target, dtype: float64
```

```
heart_data['target'].value_counts(normalize=True).plot(kind='bar')
```

```
<Axes: >
```

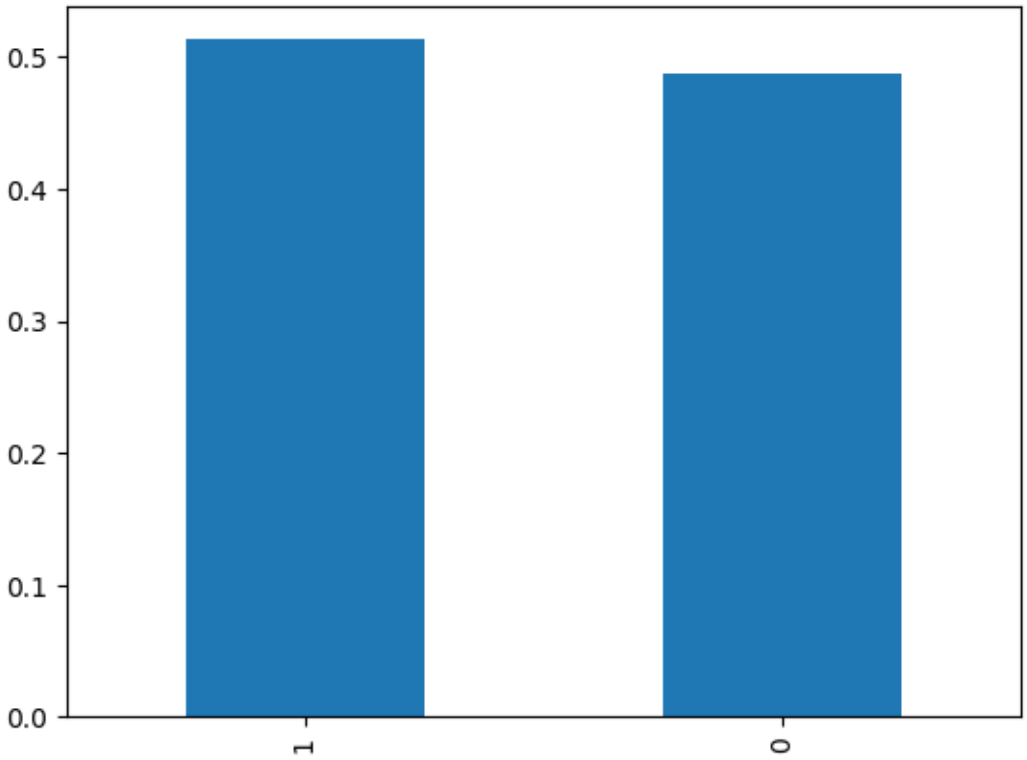

Figure 6.1: Bar-plot of diseased and healthy data

```
heart_data['Gender'] = heart_data['sex'].replace([1.0, 0.0], ['male', 'female'])
heart_data.groupby('Gender')['target'].mean()
```

```
Gender
female    0.724
male      0.421
Name: target, dtype: float64
```

```
sns.catplot(data = heart_data, y='target', x = 'Gender', kind='bar')
plt.show()
```

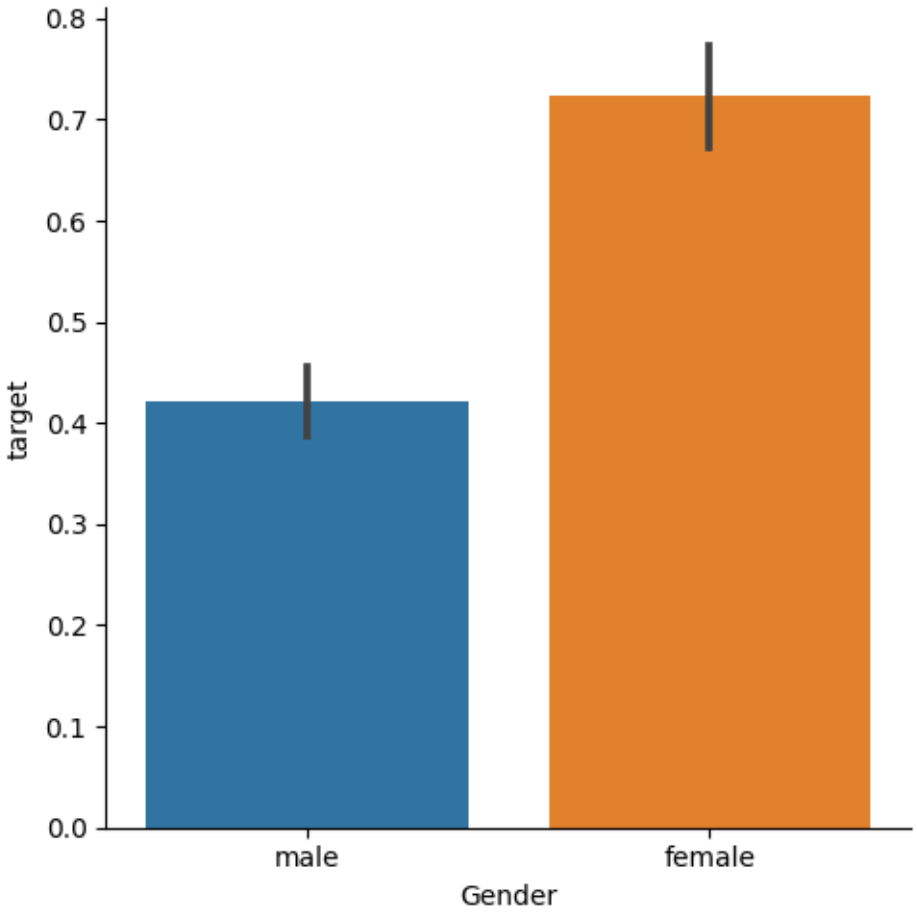

Figure 6.2: Bar-plot of diseased data based on gender

```
plt.figure(figsize=(12,7))
sns.scatterplot(data = heart_data, y='chol', x = 'age', hue = 'target')
plt.show()
```

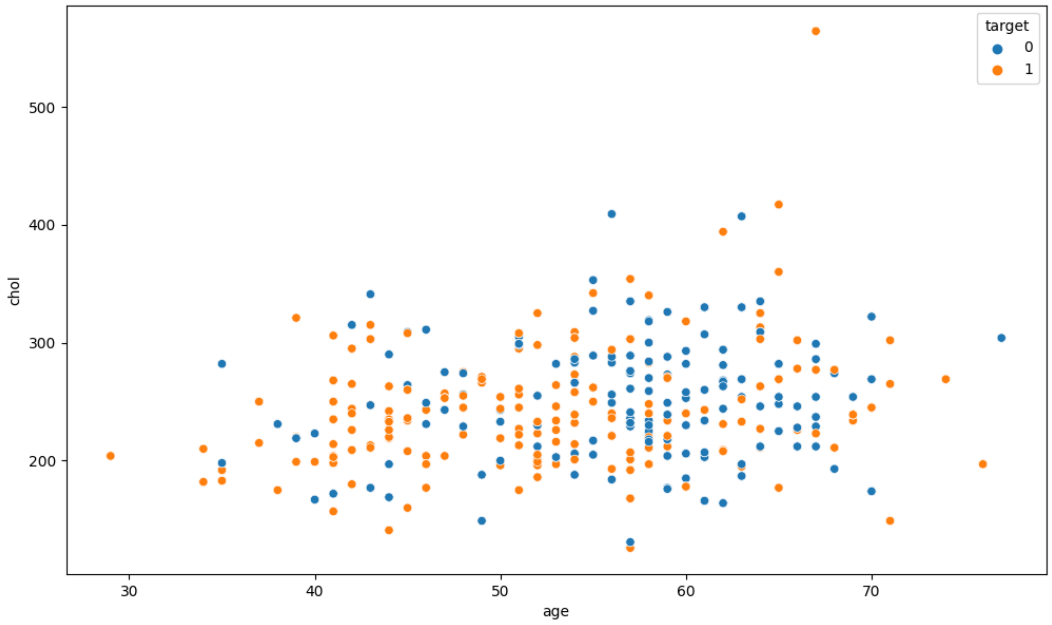

Figure 6.3: Scatter-plot of diseased data on age and chol

6.3 DATA EXPLORATION

```
plt.figure(figsize=(12,7))
sns.heatmap(heart_data.corr(),linewidth=.01,annot=True,cmap="winter")
plt.show()
```

```
<ipython-input-27-981f1a1d202f>:2: FutureWarning: The default value of numeric_only i
  sns.heatmap(heart_data.corr(),linewidth=.01,annot=True,cmap="winter")
```

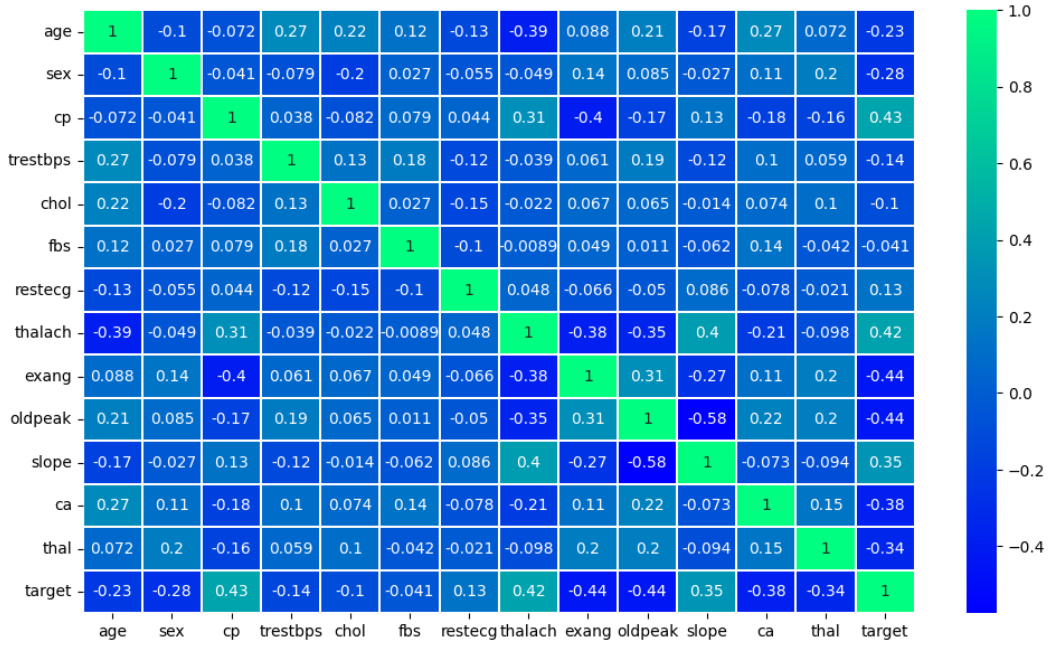

Figure 6.4: Co-relation Heat-map of all the attributes

```
heart_data.hist(figsize=(12,12))
```

```
array([[<Axes: title={'center': 'age'}>, <Axes: title={'center': 'sex'}>,
        <Axes: title={'center': 'cp'}>,
        <Axes: title={'center': 'trestbps'}>],
       [<Axes: title={'center': 'chol'}>,
        <Axes: title={'center': 'fbs'}>,
        <Axes: title={'center': 'restecg'}>,
        <Axes: title={'center': 'thalach'}>],
       [<Axes: title={'center': 'exang'}>,
        <Axes: title={'center': 'oldpeak'}>,
        <Axes: title={'center': 'slope'}>,
        <Axes: title={'center': 'ca'}>],
       [<Axes: title={'center': 'thal'}>,
        <Axes: title={'center': 'target'}>, <Axes: >, <Axes: >]],
      dtype=object)
```

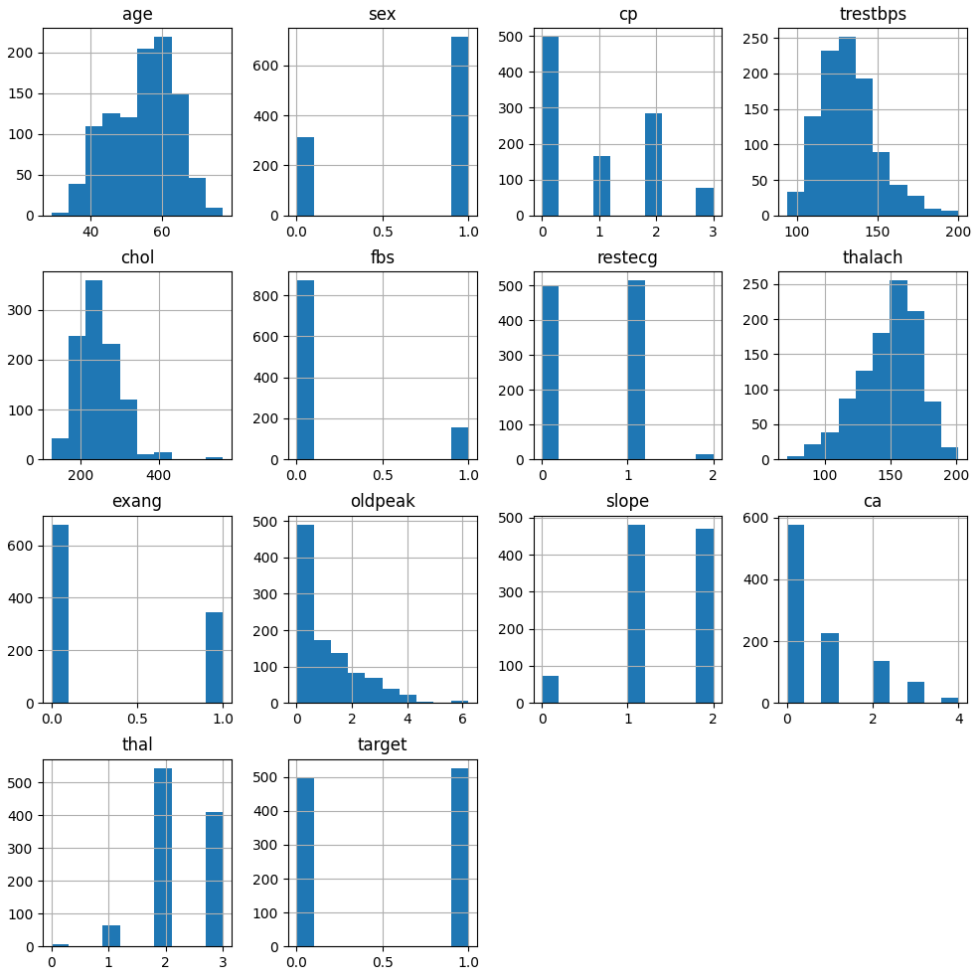

Figure 6.5: Histogram of all the attributes

```
heart_data = heart_data.drop(['Gender'], axis=1)
# creating a subset of categorical varibles that are labeled as float64
col_int = heart_data.nunique().reset_index()
col_int
```

```
       index    0
0        age   41
1        sex    2
2         cp    4
3   trestbps   49
4       chol  152
5        fbs    2
6    restecg    3
7    thalach   91
8      exang    2
9    oldpeak   40
10     slope    3
11        ca    5
12      thal    4
13    target    2
```

```
# changing all datatypes to integer
col_int.columns=['features','categories']
col_int['categories'] = col_int['categories'].astype('int64')

## sort columns based on the number of unique values
col_int = col_int.sort_values(by='categories')
col_int = col_int[col_int.categories<10]
col_int.features.values
```

```
array(['sex', 'fbs', 'exang', 'target', 'restecg', 'slope', 'cp', 'thal',
       'ca'], dtype=object)
```

```
fr_cat = heart_data[col_int.features.values]
fr_cat
```

	sex	fbs	exang	target	restecg	slope	cp	thal	ca
0	1	0	0	0	1	2	0	3	2
1	1	1	1	0	0	0	0	3	0
2	1	0	1	0	1	0	0	3	0
3	1	0	0	0	1	2	0	3	1
4	0	1	0	0	1	1	0	2	3

```
...   ...  ...    ...     ...       ...     ...  ..   ...  ..
1020    1    0      1       1         1       2    1    2    0
1021    1    0      1       0         0       1    0    3    1
1022    1    0      1       0         0       1    0    2    1
1023    0    0      0       1         0       2    0    2    0
1024    1    0      0       0         1       1    0    3    1

[1025 rows x 9 columns]
```

```
plt.figure(figsize=(30,20))
for i in enumerate(fr_cat.columns):
  plt.subplot(3, 5, i[0]+1)
  sns.countplot(x=i[1], hue='target', data = fr_cat)
```

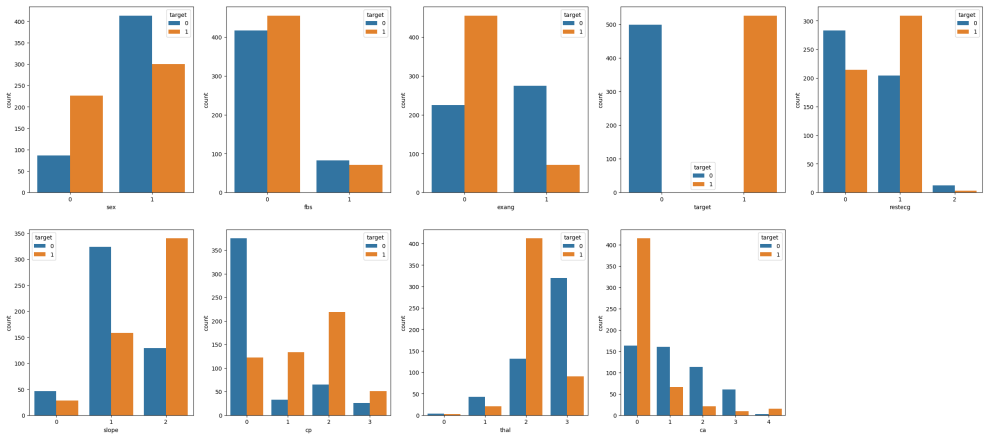

Figure 6.6: Count-plot of diseased on categorical data

```
## filterout a subset of categorical variable
fr_cont = heart_data.select_dtypes(include=['float'])
fr_cont.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 1 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   oldpeak  1025 non-null   float64
dtypes: float64(1)
memory usage: 8.1 KB
```

```
# display subset of integer type categorical variable
fr_cont = heart_data.drop(fr_cat, axis=1)
fr_cont.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1025 non-null   int64
 1   trestbps  1025 non-null   int64
 2   chol      1025 non-null   int64
 3   thalach   1025 non-null   int64
 4   oldpeak   1025 non-null   float64
dtypes: float64(1), int64(4)
memory usage: 40.2 KB
```

```
fr_cont.corr()
```

	age	trestbps	chol	thalach	oldpeak
age	1.000	0.271	0.220	-0.390	0.208
trestbps	0.271	1.000	0.128	-0.039	0.187
chol	0.220	0.128	1.000	-0.022	0.065
thalach	-0.390	-0.039	-0.022	1.000	-0.350
oldpeak	0.208	0.187	0.065	-0.350	1.000

```
plt.figure(figsize=(12,7))
sns.heatmap(fr_cont.corr())
plt.show()
```

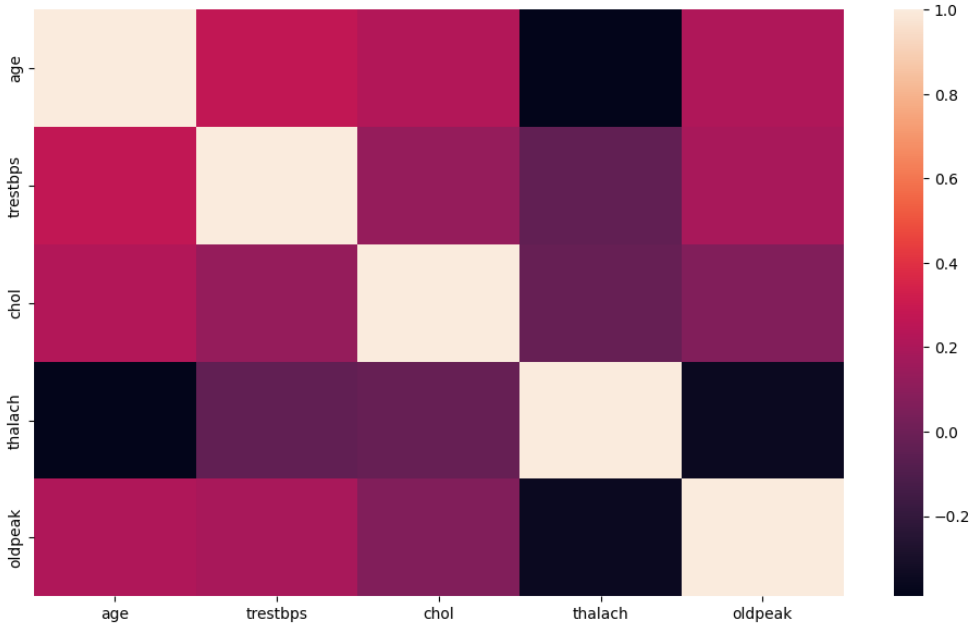

Figure 6.7: Heat-map of co-relation on continuous data

```
plt.figure(figsize=(12,7))
fr_cont.boxplot(grid=False)
plt.show()
```

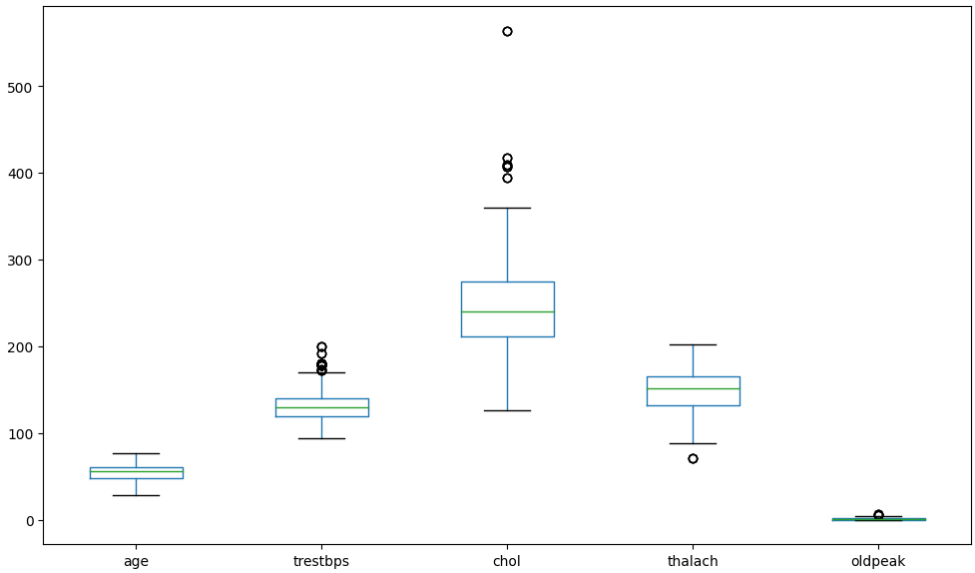

Figure 6.8: Box-plot of all continuous data

```
g = sns.FacetGrid(heart_data, row='sex', col='cp')
g.map(sns.histplot, 'target')
plt.show()
```

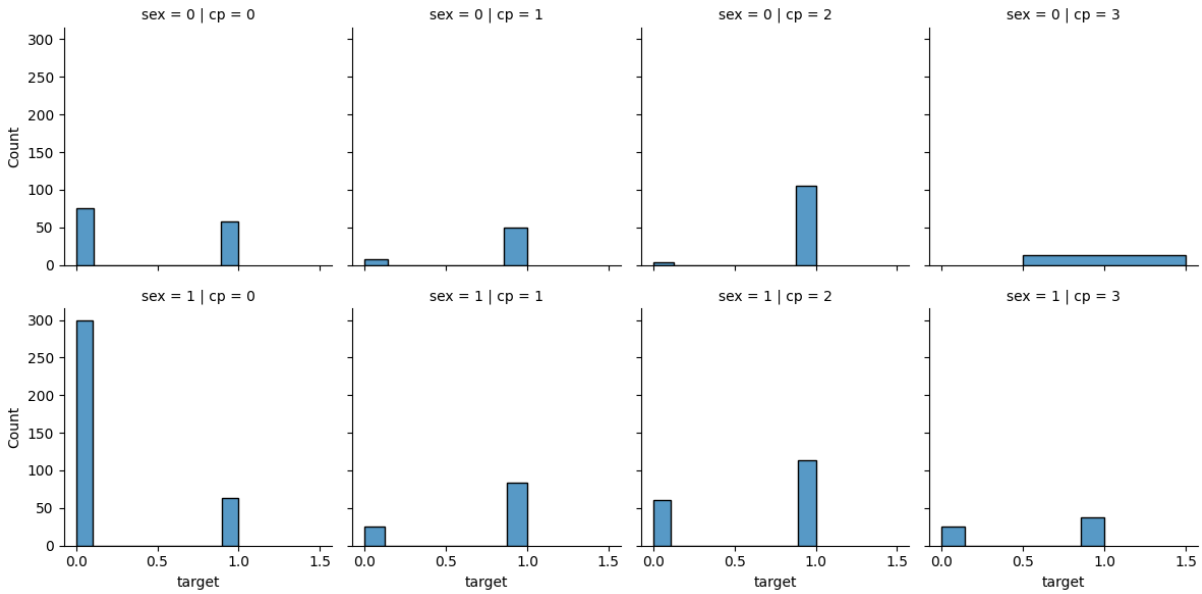

Figure 6.9: Hist-map of diseased on sex and chest-pain

```
g = sns.FacetGrid(heart_data, row='sex', col='target', height=5)
g.map(sns.scatterplot, 'age','chol')
plt.show()
```

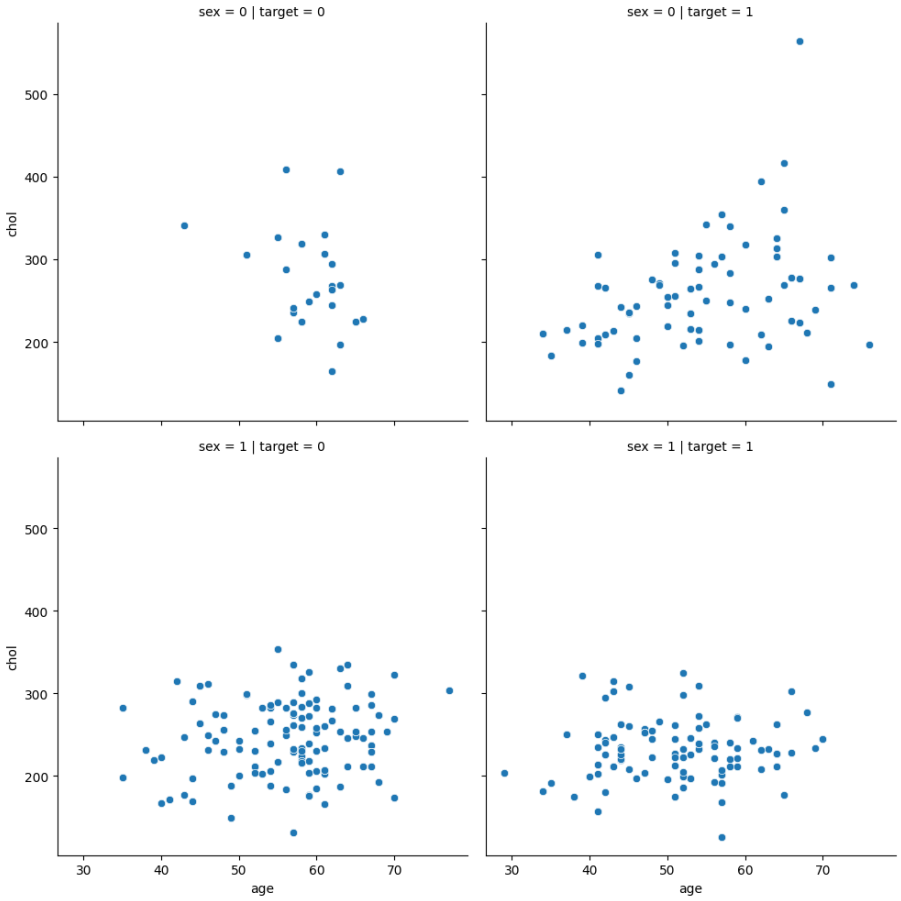

Figure 6.10: Scatter-plot on age and chol of facetgrid on sex and disesed

```
fig, ax = plt.subplots(figsize=(12,7))
sns.boxplot(data=heart_data, x="sex", y='chol', ax=ax)
plt.show()
```

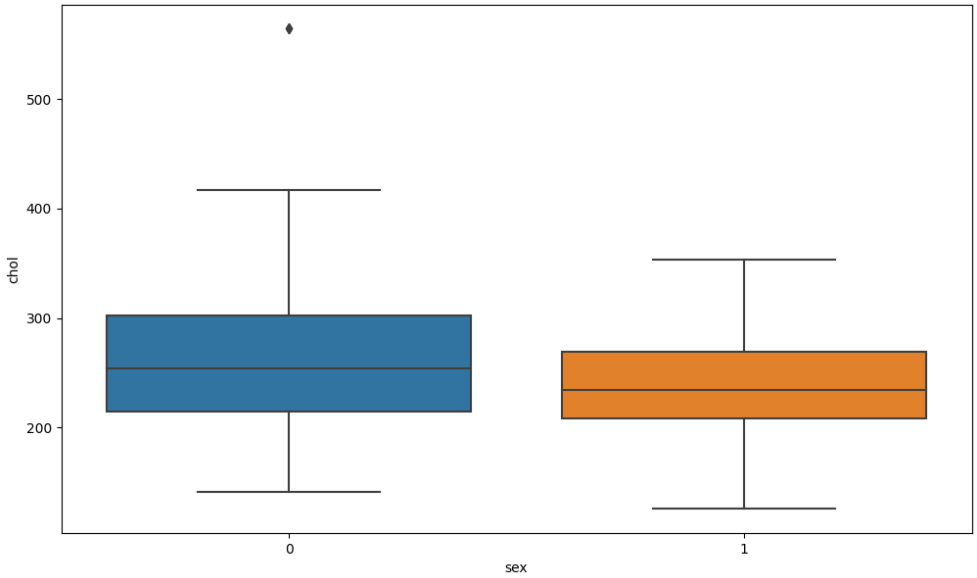

Figure 6.11: Box-plot of sex and chol

```
plt.figure(figsize=(30,20))
for i in enumerate(fr_cont.columns):
  plt.subplot(6, 4, i[0]+1)
  sns.boxplot(x=i[1], data = fr_cont)
plt.tight_layout()
```

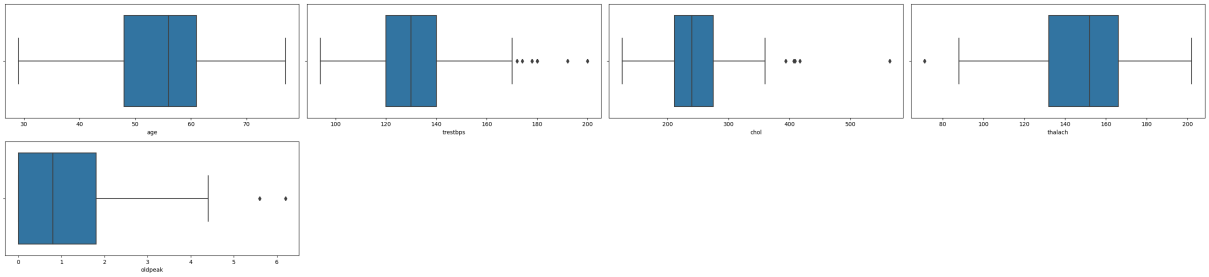

Figure 6.12: Box-plot of all continuous data

```
%matplotlib inline
```

```
pd.crosstab(heart_data['sex'],heart_data['target']).plot.bar(stacked=True)
```

```
<Axes: xlabel='sex'>
```

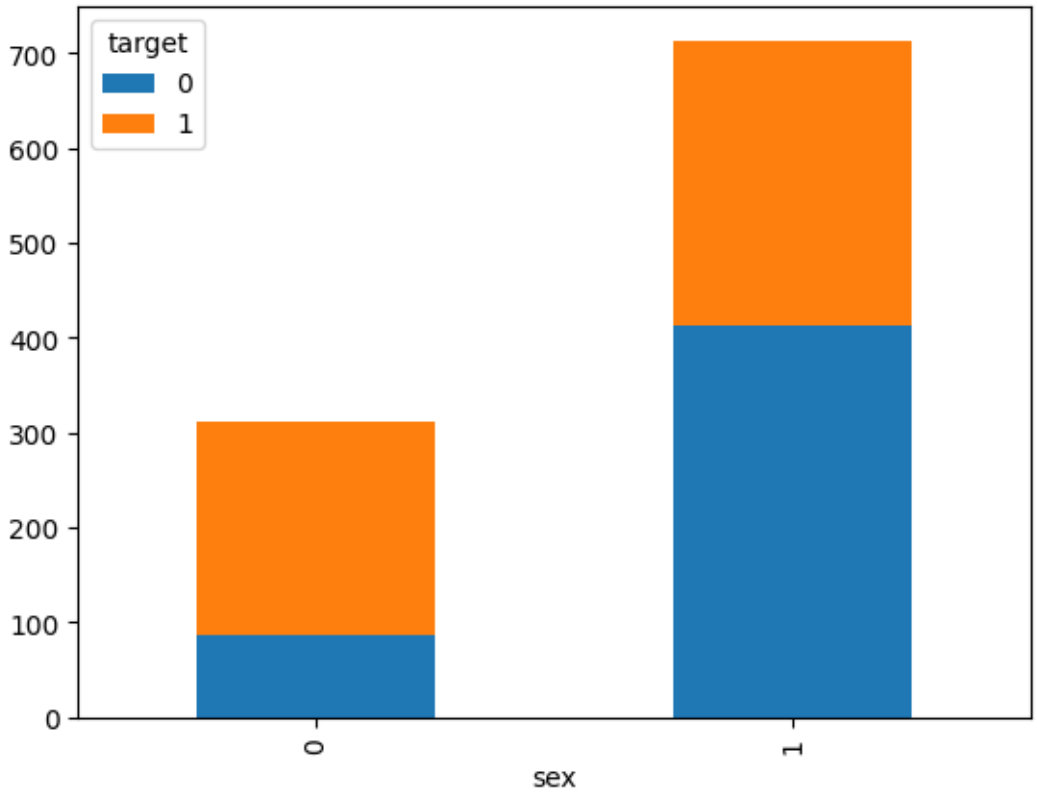

Figure 6.13: Plot-bar of diseased on sex

```
sns.catplot(x='sex',y='age',hue='target',kind='box',data=heart_data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7cc6cb5f1330>
```

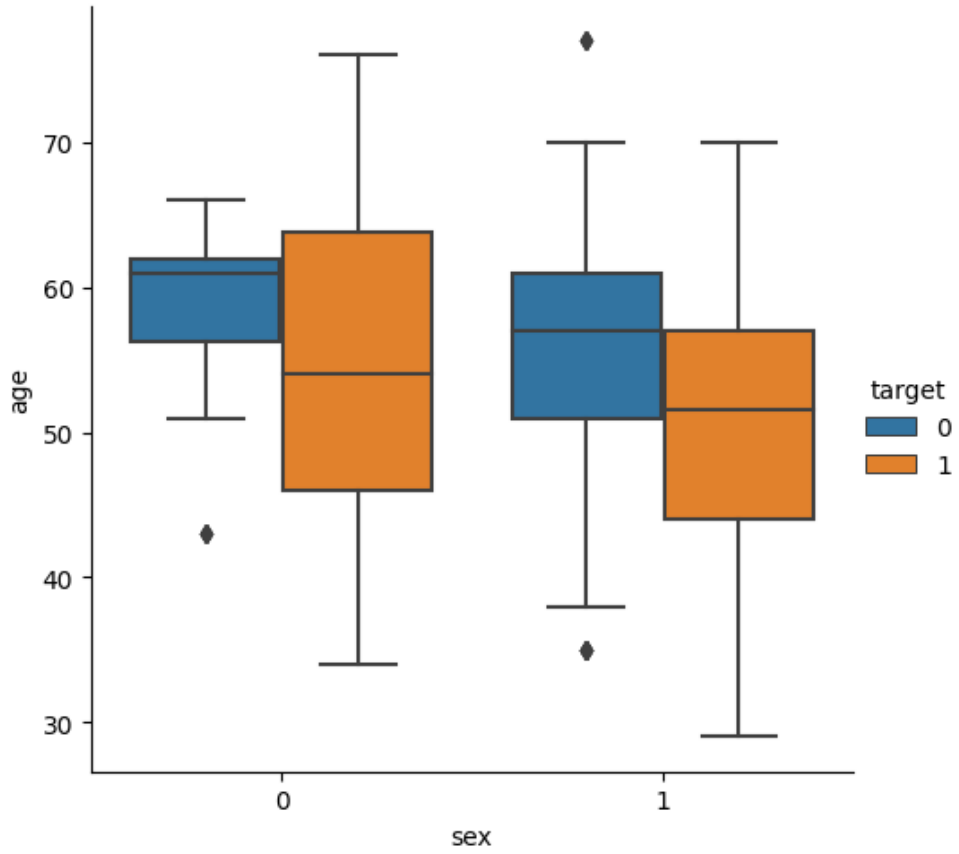

Figure 6.14: Cat-plot of diseased on sex and age

6.4 GENERATE MACHINE LEARNING MODEL

```
## distribution in target varible
heart_data['target'].value_counts(normalize=True)
```

```
1    0.513
0    0.487
Name: target, dtype: float64
```

1 -> Defective Heart 0 -> Healthy Heart

```
heart_data['target'].value_counts().plot(kind='bar')
```

```
<Axes: >
```

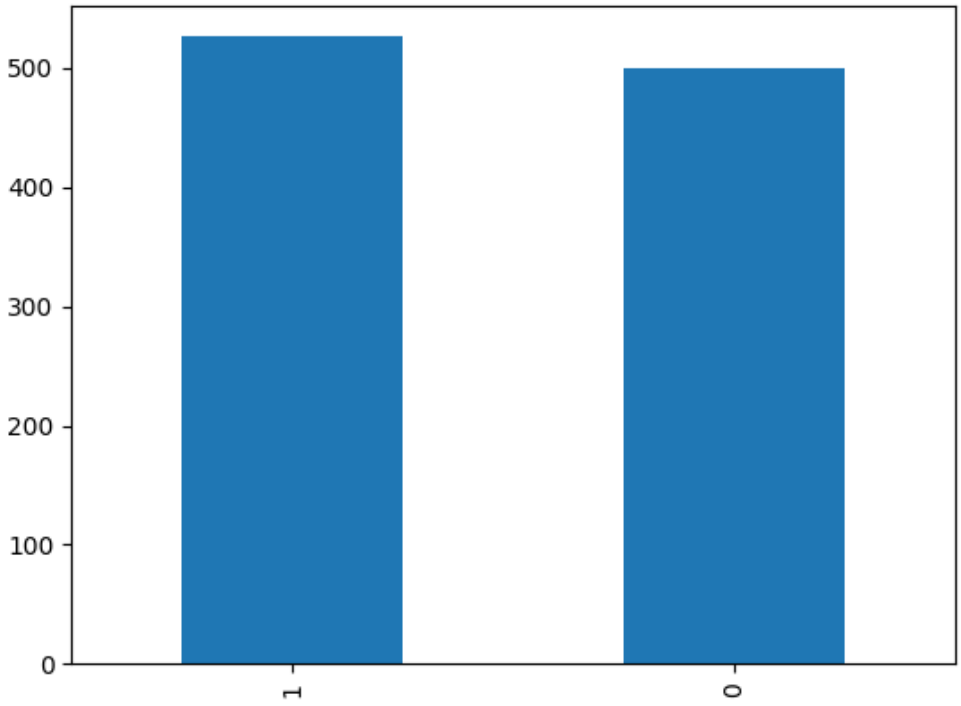

Figure 6.15: Bar-plot of diseased

```
# visualizing distribution in pi-chart
plt.figure(figsize=(10,7))
heart_data['target'].value_counts().plot(kind='pie', autopct='%1.1f', labels=['No Hea
plt.show()
```

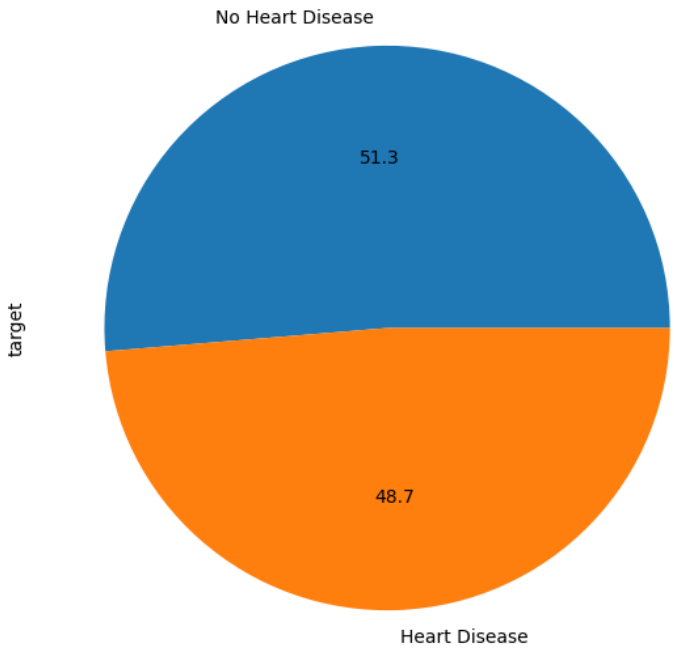

Figure 6.16: Pi-chart of diseased

6.4.1 SPLITTING THE DATA FOR MODEL IMPLEMENTATION

```
X = heart_data.drop(columns='target', axis=1)
y = heart_data['target']
```

```
X.shape
```

```
(1025, 13)
```

```
print(X)
```

```
      age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0      52    1   0       125   212    0        1      168      0    1.000
1      53    1   0       140   203    1        0      155      1    3.100
2      70    1   0       145   174    0        1      125      1    2.600
3      61    1   0       148   203    0        1      161      0    0.000
4      62    0   0       138   294    1        1      106      0    1.900
...   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
1020   59    1   1       140   221    0        1      164      1    0.000
1021   60    1   0       125   258    0        0      141      1    2.800
1022   47    1   0       110   275    0        0      118      1    1.000
1023   50    0   0       110   254    0        0      159      0    0.000
1024   54    1   0       120   188    0        1      113      0    1.400

      slope  ca  thal
0         2   2     3
1         0   0     3
2         0   0     3
3         2   1     3
4         1   3     2
...     ...  ..   ...
1020      2   0     2
1021      1   1     3
1022      1   1     2
1023      2   0     2
1024      1   1     3

[1025 rows x 13 columns]
```

```
y.shape
```

```
(1025,)
```

```
print(y)
```

```
0       0
1       0
2       0
3       0
4       0
       ..
1020    1
1021    0
1022    0
1023    1
1024    0
Name: target, Length: 1025, dtype: int64
```

Splitting the dataset into training data and test data

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(1025, 13) (820, 13) (205, 13)
```

MOdel Training

Logistic Regression

```
model = LogisticRegression()
```

```
# training the logisticRegression model with training data
model.fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: Conver
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
LogisticRegression()
```

Model Evaluation

Accuracy Score

```
# accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, y_train)

print("Accuracy on Training Data :", training_data_accuracy*100)
```

```
Accuracy on Training Data : 85.73170731707317
```

6.4.2 LOGISTIC REGRESSION MODEL

```
# accuracy on testing data
X_test_prediction = model.predict(X_test)
testing_data_accuracy = accuracy_score(X_test_prediction, y_test)

print("Accuracy on Testing Data :", testing_data_accuracy*100)
```

```
Accuracy on Testing Data : 83.41463414634146
```

```
from sklearn.model_selection import cross_val_score, GridSearchCV
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression(C=1.0, class_weight='balanced', dual=False,
                      fit_intercept=True, intercept_scaling=1, l1_ratio=None,
                      max_iter=100, multi_class='auto', n_jobs=None, penalty='l2',
                      random_state=1234, solver='lbfgs', tol=0.0001, verbose=0,
                      warm_start=False)
model1=lr.fit(X_train,y_train)
prediction1=model1.predict(X_test)
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,prediction1)
cm
sns.heatmap(cm, annot=True,cmap='winter',linewidths=0.3, linecolor='black',annot_kws=
TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]

print('Testing Accuracy for Logistic Regression:',(TP+TN)/(TP+TN+FN+FP))
print('Testing Sensitivity for Logistic Regression:',(TP/(TP+FN)))
print('Testing Specificity for Logistic Regression:',(TN/(TN+FP)))
print('Testing Precision for Logistic Regression:',(TP/(TP+FP)))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: Conver
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

6.5 RESULTS

```
Testing Accuracy for Logistic Regression: 0.8195121951219512
Testing Sensitivity for Logistic Regression: 0.851063829787234
Testing Specificity for Logistic Regression: 0.7927927927927928
Testing Precision for Logistic Regression: 0.7766990291262136
```

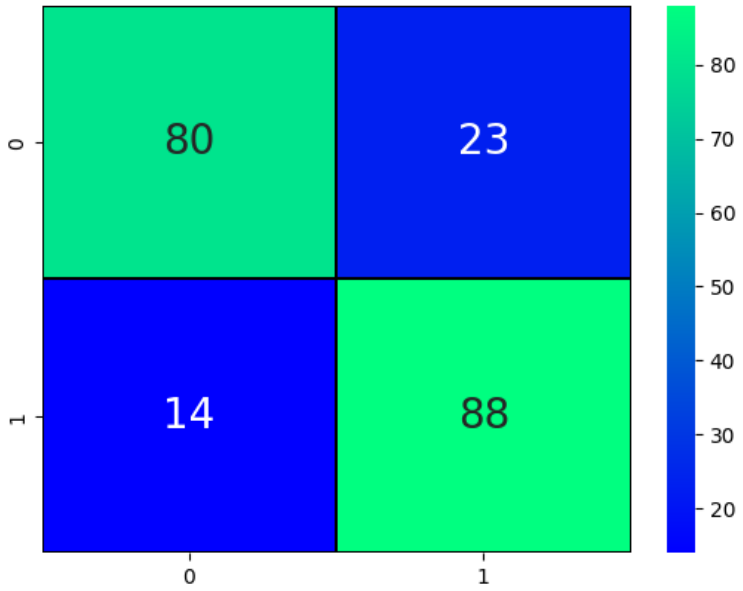

Figure 6.17: Heat-map of result

```
from sklearn.metrics import classification_report
print(classification_report(y_test, prediction1))
```

	precision	recall	f1-score	support
0	0.85	0.78	0.81	103
1	0.79	0.86	0.83	102
accuracy			0.82	205
macro avg	0.82	0.82	0.82	205
weighted avg	0.82	0.82	0.82	205

6.6 TESTING THE PREDICTION

```
# for normal patient
input=(58,1,0,114,318,0,2,140,0,4.4,0,3,1)
input_as_numpy=np.asarray(input)
input_reshaped=input_as_numpy.reshape(1,-1)
pre1=lr.predict(input_reshaped)
if(pre1==1):
  print("The patient seems to be have heart disease:(")
else:
  print("The patient seems to be Normal:)")
The patient seems to be Normal:)
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not
  warnings.warn(
# for diseased patient
input=(50,0,1,120,244,0,1,162,0,1.1,2,0,2)
input_as_numpy=np.asarray(input)
input_reshaped=input_as_numpy.reshape(1,-1)
pre1=tree_model.predict(input_reshaped)
if(pre1==1):
  print("The patient seems to be have heart disease:(")
else:
  print("The patient seems to be Normal:)")
The patient seems to be have heart disease:(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not
  warnings.warn(
```

Chapter 7

CONCLUSION,LIMITATIONS AND FUTURE SCOPE

7.1 CONCLUSION:

The heart disease prediction model has emerged as a pivotal tool in modern healthcare, offering a glimpse into the potential for predictive analytics to transform patient care. Through the amalgamation of advanced machine learning algorithms and extensive datasets, this model exhibits a profound ability to forecast the likelihood of heart disease onset, enabling early intervention and personalized treatment strategies.

At its core, this predictive model harnesses the power of data. It ingests and analyzes a multitude of variables encompassing demographics, medical history, lifestyle choices, and diagnostic tests. By meticulously scrutinizing these inputs, the model discerns intricate patterns and correlations, identifying key risk factors that contribute to cardiovascular ailments.

The implications of this predictive model are multifaceted and profound. Primarily, it empowers healthcare providers with proactive insights, allowing for timely intervention and preventive measures. Identifying high-risk individuals at an early stage permits targeted interventions, whether through lifestyle modifications, medication, or specialized monitoring, potentially averting or mitigating the progression of heart disease. This preemptive approach not only improves patient outcomes but also alleviates the strain on healthcare systems by reducing hospitalizations and costly treatments associated with advanced stages of cardiovascular conditions.

In conclusion, the heart disease prediction model stands as a beacon of innovation in healthcare, showcasing the immense potential of data-driven approaches in transforming patient care. Its ability to forecast heart disease risk not only revolutionizes early intervention strategies but also fosters a paradigm shift towards personalized and preventive medicine. As advancements continue and ethical concerns are addressed, this model is poised to redefine the landscape of cardiovascular healthcare, ultimately saving lives and improving the quality of care for countless individuals worldwide.

7.2 LIMITATIONS:

Heart disease prediction models are valuable tools for assessing the risk of cardiovascular issues, yet they do have limitations. These limitations include:

1. **Data Quality:** The accuracy of these models heavily relies on the quality and quantity of the data used for training. If the data is incomplete, biased, or not representative of diverse populations, the model's predictions may lack generalizability.

2. **Complexity of Risk Factors:** Heart disease is multifactorial, influenced by numerous variables like genetics, lifestyle, environmental factors, and medical history. Prediction models might not encompass all these elements, leading to oversimplification and potentially missing crucial risk factors.

3. **Evolution of Risk Factors:** Risk factors for heart disease can evolve over time, influenced by changing lifestyles, new medical findings, or societal shifts. Models might not be updated frequently enough to reflect these changes, affecting their accuracy.

4. **Limited Scope:** Models might focus on specific aspects of heart disease, such as predicting a cardiac event or assessing the risk of coronary artery disease, while overlooking other related conditions or nuances in disease progression.

5. **Interpretability and Trust:** Complex machine learning models often lack interpretability, making it challenging for healthcare professionals to understand the rationale behind predictions. This can impact their trust in the model's recommendations.

6. **Ethical and Privacy Concerns:** Using sensitive health data for prediction raises ethical concerns regarding data privacy, consent, and potential biases in decision-making.

7. **Population Variability:** Models trained on a particular population might not generalize well to other demographics or ethnic groups due to variations in genetic predispositions, lifestyle, or healthcare disparities.

8. **Overfitting and Validation:** Models might perform exceptionally well on the data they were trained on but could struggle with new, unseen data, indicating

overfitting and reduced generalizability.

7.3 FUTURE SCOPE:

The future of heart disease prediction models holds immense promise in transforming healthcare. With advancements in technology and data analytics, these models are poised to revolutionize preventive medicine, diagnosis, and personalized treatment plans.

Machine learning algorithms and artificial intelligence have shown remarkable potential in analyzing vast amounts of patient data, including medical history, genetic information, lifestyle factors, and diagnostic tests. These models can identify patterns, correlations, and risk factors that might not be immediately apparent to healthcare providers, thereby enabling earlier and more accurate predictions of heart disease.

The integration of wearable devices and continuous health monitoring will further enhance these models by providing real-time data streams. This allows for constant monitoring of vital signs, activity levels, and other relevant metrics, leading to more precise risk assessments and timely interventions.

Moreover, the future of heart disease prediction models lies in their ability to facilitate personalized medicine. By considering individual variations in genetics, lifestyle, and environmental factors, these models can tailor prevention strategies and treatment plans specifically for each patient, maximizing effectiveness and minimizing adverse outcomes.

Ethical considerations, data privacy, and the need for transparent and interpretable models remain critical challenges. However, ongoing research and collaborations between healthcare professionals, data scientists, and regulatory bodies aim to address these concerns while advancing the accuracy and reliability of predictive models.

In essence, the future of heart disease prediction models is poised to significantly impact healthcare by enabling proactive interventions, personalized care, and improved patient outcomes, ultimately contributing to a healthier global population.

REFERENCES

[1] L. Caruccio, S. Cirillo, G. Polese, G. Solimando, S. Sundaramurthy, and G. Tortora, "Can chatgpt provide intelligent diagnoses? a comparative study between predictive models and chatgpt to define a new medical diagnostic bot," *Expert Systems with Applications*, vol. 235, p. 121186, 2024.

[2] D. Parkhi, N. Periyathambi, Y. Ghebremichael-Weldeselassie, V. Patel, N. Sukumar, R. Siddharthan, L. Narlikar, and P. Saravanan, "Prediction of postpartum prediabetes by machine learning methods in women with gestational diabetes mellitus," *Iscience*, vol. 26, no. 10, 2023.

[3] R. Sree Vidya, K. Nandhini, R. Mathu Shri, and S. Shanmuga Priyanka Devi, "Heart disease prognosis using artificial intelligence," 2021.

[4] K. Amen, M. Zohdy, and M. Mahmoud, "Machine learning for multiple stage heart disease prediction," in *Proceedings of the 7th International Conference on Computer Science, Engineering and Information Technology*, pp. 205–223, 2020.